Middleware - comparatie intre comunicare sincrona prin obiecte distribuite si comunicare asincrona prin mesaje

Tehnologii middleware bazate pe mesaje

Sisteme software complexe sunt realizate cu ajutorul tehnologiilor middleware ,acestea fiind folosite ca si elemente de legatura pentru interconectarea de sisteme software independente. Nu exista limitare din punct de vedere thnologic in ceea ce priveste dezvolatrea de sisteme software care sunt interconectate prin tehnologii middleware si pot rula pe diferite platforme hardware si/sau software.

Tehnologiile middleware folosite la interconectarea de sisteme prezinta o adaptare usoara in sensul in care nu intervin modificari complexe.

Principiul comunicarii asincrone prin mesaje se bazeaza pe comunicarea dintre oameni. Paradigma serviciilor de retea se bazeaza pe schimbul de mesaje intre procese.

In cazul comunicarii prin obiecte distribuite, cererea trebuie sa fie insotita de un raspuns imediat.

Comunicarea prin mesaje care este data-oriented in timp ce comunicarea prin obiecte distribuite este action-oriented ceea ce inseamna ca este concentrata pe invocarea de operatii, in timp ce trimiterea de date trece in plan secund. Comunicarea prin obiecte distribuite este mult mai naturala pentru dezvoltarea software orintata pe obiecte.

Principiile de functioare ale celor doua metode de comunicare

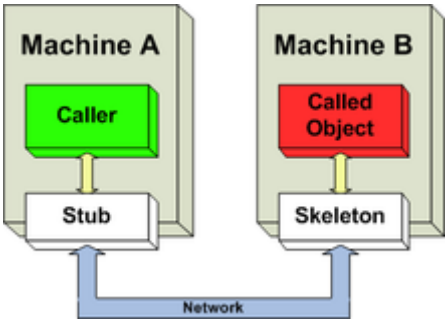

Fig.Comunicarea sincrona prin obiecte distribuite

Obiectele distribuite(in sensul programarii orientate pe obiecte) sunt repartizate la anumite spatii de adrese, fie pe aceeasi unitate sau in mai multe unitati conectate prin retea dar care lucreaza impreuna impartasind informatie si invocand metode.

Comunicarea prin obiecte distribuite are ca si scop permiterea obiectelor sa acceseze informatie si sa invoce metode ale obiectelor. Invocarea unei metode a unui obiect distribuit se numeste remote method invocation(RMI).

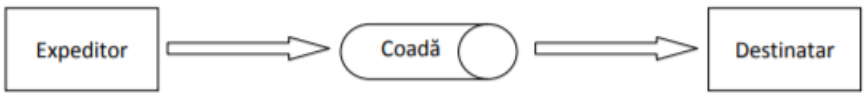

Fig. Comunicarea asincrona prin mesaje

Termenul de “messaging” se refera la o metoda de comunicare intre doua aplicatii sau intre doua parti ale unei aplicatii distribuite: un producãtor si un consumator de mesaje.

Intre producator si consumator exista un agent intermediar ("Message Broker"), numit si “Provider” sau MOM (“message-oriented middleware”), care asigurã memorarea temporara a mesajelor, transformarea, distribuirea (rutarea) catre mai multi receptori, filtrarea mesajelor pe baza proprietatilor sau continutului, transformarea mesajelor, persistenta mesajelor si alte functii.

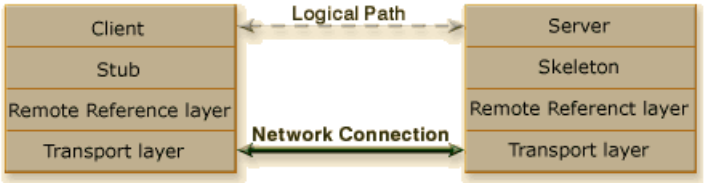

Fig. Relatia Client-Server

Remote Method Invocation (RMI)

1. Comunicare sincrona - este asteptat un raspuns in urma unei cereri. Pentru a realiza acest lucru, serviciul RMI trebuie sa fie activ la momentul comunicarii.
2. Cuplare bine realizata - sistemul de cerere este cuplat la interfata declarata in serviciul RMI
3. Comunicatie nedemna de incredere - daca se intampla ceva in timpula in care comunicatia se produce ,se va pierde cererea si vor trebui rezolvate le nivel aplicatie.

Avantaje ale utilizarii:

- Usor de a incepe prin a scrie o interfata si implementarea pentru aceasta.

Dezavantaje ale utilizarii:

- Poate utiliza doar platforme utilizate de Java.
- Nu se poate folosi cod in afara sferei Java.
- Probleme de securitate trebuie monitorizate mai atent.

Folosire RMI

- Daca avem un proces care necesita putere de calcul mare si dorim mai multa putere de calcul pentru imbunatatirea perfermantei ,atunci putem folosi RMI deoarece va oferi putere de calcul pe apeluri de metode simple.

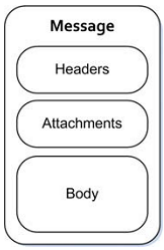

Fig. Format Mesaje

Java Message Service (JMS)

Primele produse MOM (MQSeries de la IBM) si-au dovedit utilitatea dar si necesitatea de standarde. Un astfel de standard este JMS, care este un API Java (interfete si clase) pentru producatori si receptori de mesaje.

1. Comunicatie asincrona - Cel care initiaza cererea nu asteapta un raspuns ci trimite mesajul si apoi inchide conexiunea.Nu este nevoie de serviciu de asistenta la momentul comunicarii.
2. Cuplare slaba - Cuplarea nu este realizata bine ca si in cazul RMI. Se poate folosi orice format de data pentru comunicarea prin cozi si topicuri (ex. XML,JSON)
3. Comunicatie demna de incredere - Aceasta nu este ca si RMI pentru ca are un broker de mesaje intre sisteme si broker-ul de mesaje are capacitatea de tratare a erorilor si persistenta mesajelor si le inmaneaza partii responsabile.

Avantaje:
- Datorita comunicarii asincrone, nu toate modulele trebuie sa fie active pentru ca aplicatia sa functioneze ca intreg.

Dezavantaje:
- Cost suplimentar pentru configurarea si intretinerea broker-ului de mesaje.

Folosire JMS
- Daca avem vreun proces care va primii un request destul de mare si care poate sa functioneze asincron, atunci am putea opta pentru implementarile cozii JMS si sa le rulam separat si sa comunice prin coada.Urmatorul aspect este daca avem un proces care trebuie sa transmita informatie catre multe aplicatii in aval, atunci am putea opta pentru implementarea JMS Topic.

Bibliografie:

1. http://www.aut.upt.ro/staff/diercan/data/PSSC/curs-05.pdf

2. http://control.aut.utcluj.ro/scd/lab4/distributedObjects.htm

3. https://www.infor.uva.es/~jjalvarez/asignaturas/SD/lectures/distributed_objects.pdf